

Pressure segregation methods based on a discrete pressure Poisson equation. An algebraic approach

Santiago Badia^{*,†} and Ramon Codina

CIMNE, Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain

SUMMARY

In this paper, we introduce some pressure segregation methods obtained from a non-standard version of the discrete monolithic system, where the continuity equation has been replaced by a pressure Poisson equation obtained at the discrete level. In these methods it is the velocity instead of the pressure the extrapolated unknown. Moreover, predictor–corrector schemes are suggested, again motivated by the new monolithic system. Key implementation aspects are discussed, and a complete stability analysis is performed. We end with a set of numerical examples in order to compare these methods with classical pressure-correction schemes. Copyright © 2007 John Wiley & Sons, Ltd.

Received 30 October 2006; Revised 9 April 2007; Accepted 20 April 2007

KEY WORDS: pressure segregation; velocity correction; predictor–corrector; incompressible Navier–Stokes equations

1. INTRODUCTION

At the continuous level, from the Navier–Stokes equations for incompressible flows (the momentum and continuity equations) a pressure Poisson equation (PPE, from now onwards) can be obtained. An alternative system to the classical Navier–Stokes equations can be the momentum equation together with the PPE instead of the continuity equation. Unfortunately, the numerical approximation of the PPE equation is an involved task due to the presence of third-order derivatives for the velocity field. In order to facilitate the numerical approximation, a *simplified* PPE could be considered by assuming that the divergence and the Laplace operators commute and deleting the viscous contribution to the PPE. Nevertheless, as it is shown by Gresho and Sani in [1], this alternative form is not equivalent to the Navier–Stokes equations. Therein, it is shown that the system with

^{*}Correspondence to: S. Badia, Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.

[†]E-mail: sbadia@cimne.upc.edu

the *simplified* PPE admits spurious solutions that are not solutions of the original Navier–Stokes equations.

We can consider an alternative version of the fully discretized monolithic system that is obtained at the discrete level. As for the continuous problem, from the discrete momentum and continuity equations a discrete pressure Poisson equation (DPPE, from now onwards) can be easily obtained. This leads to a fully discrete monolithic system composed by the DPPE and the discrete momentum equation. However, *it is not our interest to solve this DPPE monolithic problem. The aim of this paper is to design pressure segregation methods motivated by this DPPE monolithic problem.*

Pressure segregation methods for the incompressible Navier–Stokes equations have become widely popular since their appearance at the late sixties (see [2, 3]). These methods allow the uncoupling of the pressure from the velocity components, reducing sensibly the computational cost. We classify the pressure segregation methods into two families: pressure-correction (PC) methods and velocity-correction (VC) methods. The concept of PC schemes appeared when trying to obtain second-order accurate projection methods. In [4], van Kan introduced a second-order PC method where the pressure in the momentum equation was extrapolated from the value of the previous time step. In fact, the classical Chorin–Temam projection method can be understood as a PC scheme with a zero-order extrapolation of the pressure. Methods where the extracted unknown is the velocity rather than the pressure are very recent. In [5], Guermond and Shen have introduced first versions of VC methods (VC) obtained at the semi-discrete level (in time).

In this article, *we motivate VC methods at the fully discrete level from the DPPE monolithic system.* The algebraic approach allows us to avoid the discussion about the appropriate boundary conditions and motivates an alternative implementation to the one in [5]. We propose a method where the pressure is obtained from the DPPE using an extrapolated velocity. After it is computed, this pressure is used in the momentum equation in order to obtain the velocity.

The system of equations related to the DPPE method is cumbersome. This is why a further approximation is suggested in order to make the method more appealing from a computational point of view. This approximation can be understood as an improved version of the one widely used for PC methods (see [6]).

From PC methods, predictor–corrector methods can be easily obtained. Here, *we design a new predictor–corrector scheme where all the terms are motivated from the DPPE monolithic system.* This is a main difference in comparison to the existing predictor–corrector methods based on PC schemes (see [7]).

The stabilized version of VC schemes using the orthogonal subscale stabilization method (see [8]) has also been studied and some possibilities discussed.

We have proved some stability bounds for different schemes. We have compared the inherent pressure stability of VC methods with that of PC methods. This analysis is similar to that presented in [9] for PC methods. In fact, the present paper is a continuation of the effort initiated in this reference to analyse and design robust methods that uncouple the velocity and pressure calculations.

Let us list the main results of the article:

1. Design of VC methods motivated at the discrete level by the DPPE monolithic system.
2. A new implementation of VC methods.
3. Design of predictor–corrector methods motivated by the DPPE monolithic system.
4. Stabilized version of these schemes using orthogonal subscales.
5. Stability analysis for VC methods (with and without stabilization).

The outline of the paper is as follows. In Section 2, some preliminaries for the continuous and discrete problems are stated for the monolithic system. In Section 3, VC methods are introduced and discussed. Section 4 is devoted to a new family of predictor–corrector methods. The stabilized version of VC schemes is introduced in Section 5. In Section 6, some implementation aspects are discussed. Section 7 is devoted to the numerical analysis of VC methods. Stability results for different cases are obtained. The important point is that no reference at all is made to compatibility conditions between the velocity and pressure interpolating spaces, that is, they do not need to satisfy any inf–sup condition. Section 8 presents a set of numerical tests that show the behaviour of these new schemes in comparison with the classical PC methods and, finally, the paper concludes in Section 9 with some final remarks.

2. PRELIMINARIES AND PROBLEM STATEMENT

2.1. The continuous problem

Let Ω be the domain of \mathbb{R}^d occupied by the fluid, where $d=2$ or 3 is the number of space dimensions, $\Gamma=\partial\Omega$ its boundary and $[0, T]$ the time interval of analysis. The Navier–Stokes problem consists of finding a velocity \mathbf{u} and a pressure p such that

$$\partial_t \mathbf{u} - \nu \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \quad t \in (0, T) \quad (1a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad t \in (0, T) \quad (1b)$$

$$\mathbf{u} = \mathbf{u}_d \quad \text{on } \Gamma, \quad t \in (0, T) \quad (1c)$$

$$\mathbf{u} = \mathbf{u}^0 \quad \text{in } \Omega, \quad t = 0 \quad (1d)$$

where ν is the kinematic viscosity, \mathbf{f} is the force vector, \mathbf{u}_d defines the Dirichlet boundary value and \mathbf{u}^0 is the velocity initial condition.

An alternative form of the Navier–Stokes equations where the continuity equation (1b) is replaced by a new equation that replaces the mass conservation can also be considered. A PPE that implies mass conservation can be obtained by taking the divergence of the momentum equation (1a) and invoking (1c), leading to the system

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad (2a)$$

$$\Delta p = \nabla \cdot (\mathbf{f} + \nu \Delta \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u}) \quad (2b)$$

The pressure boundary condition is obtained by imposing that the normal component of the pressure gradient is equal to the normal component of the term within parenthesis in the right-hand side of (2b). As commented below, the term $\nabla \cdot (\nu \Delta \mathbf{u})$ complicates the finite element approximation of the PPE equation. We can extract this term from the equation by commuting operators, leading to a *simplified* PPE:

$$\Delta p = \nabla \cdot (\mathbf{f} - \mathbf{u} \cdot \nabla \mathbf{u}) \quad (3)$$

Unfortunately, this equation is ill-posed. It can be shown that (2a)–(3) admit extra *spurious* solutions that do not satisfy the original Navier–Stokes equations. These *spurious* solutions satisfy

$$\partial_t (\nabla \cdot \mathbf{u}) - \nu \Delta (\nabla \cdot \mathbf{u}) = 0 \quad (4)$$

which does not necessarily imply $\nabla \cdot \mathbf{u} = 0$ for $t \in (0, T)$. Thus, the false equivalence between the original continuity equation (1b) and the simplified PPE obtained at the continuous level makes (2a)–(3) an inappropriate alternative to the original Navier–Stokes equations. The problems arising when using the simplified PPE are analysed in [1].

2.2. Weak form

To pose the weak form of problem (1), we need the functional spaces $\mathcal{V}_0 = \mathbf{H}_0^1(\Omega)$ and $\mathcal{Q} = L^2(\Omega)/\mathbb{R}$. As usual, $L^2(\Omega)$ denotes the space of square integrable functions endowed with the scalar product (\cdot, \cdot) . $H^1(\Omega)$ is the space of functions in $L^2(\Omega)$ with first derivatives also in $L^2(\Omega)$, whereas $H_0^1(\Omega)$ is the subspace of $H^1(\Omega)$ of functions vanishing on $\partial\Omega$. Its dual will be denoted by $H^{-1}(\Omega)$ and the duality pairing between $H^{-1}(\Omega)$ and $H_0^1(\Omega)$ by $\langle \cdot, \cdot \rangle$. A bold character is used for the vector counterpart of these spaces.

For the evolutionary Navier–Stokes equations we also need to introduce $(\mathcal{V}_0)_t \equiv L^2(0, T; \mathcal{V}_0)$ and $\mathcal{Q}_t \equiv L^1(0, T; \mathcal{Q})$, where $L^p(0, T; X)$ is the space of time-dependent functions in a normed space X such that $\int_0^T \|f\|_X^p dt < \infty$, $1 \leq p < \infty$. Then, the weak form of (1) with homogeneous boundary conditions (for simplicity) consists of finding $\mathbf{u} \in (\mathcal{V}_0)_t$ and $p \in \mathcal{Q}_t$ such that:

$$(\partial_t \mathbf{u}, \mathbf{v}) + \nu(\nabla \mathbf{u}, \nabla \mathbf{v}) + (\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v}) + \langle \nabla p, \mathbf{v} \rangle = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{V}_0 \quad (5a)$$

$$\langle \nabla q, \mathbf{u} \rangle = 0 \quad \forall q \in \mathcal{Q} \quad (5b)$$

The symbol (\cdot, \cdot) has been used also for the convective term even though $\mathbf{u} \cdot \nabla \mathbf{u}$ does not belong to $L^2(\Omega)$. In these equations, we have assumed that the force vector belongs to $\mathbf{H}^{-1}(\Omega)$.

If we want to obtain the weak form of (2) with homogeneous Dirichlet boundary conditions, we need some more regularity in the solution. Let us consider the functional spaces $(\tilde{\mathcal{V}}_0)_t \equiv L^2(0, T; \tilde{\mathcal{V}}_0)$ and $\tilde{\mathcal{Q}}_t \equiv L^2(0, T; \tilde{\mathcal{Q}})$, with $\tilde{\mathcal{V}}_0 \equiv \mathbf{H}_0^1(\Omega) \cap \mathbf{H}^2(\Omega)$ and $\tilde{\mathcal{Q}} \equiv H^1(\Omega)/\mathbb{R}$. Now, we assume that the force vector belongs to $\mathbf{L}^2(\Omega)$. The weak form consists of finding $[\mathbf{u}, p] \in (\tilde{\mathcal{V}}_0)_t \times (\tilde{\mathcal{Q}})_t$ such that

$$(\partial_t \mathbf{u}, \mathbf{v}) + \nu(\nabla \mathbf{u}, \nabla \mathbf{v}) + (\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v}) + (\nabla p, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad (6a)$$

$$(\nabla p, \nabla q) = (\mathbf{f} - \mathbf{u} \cdot \nabla \mathbf{u} + \nu \Delta \mathbf{u}, \nabla q) \quad (6b)$$

for all $[\mathbf{v}, q] \in \tilde{\mathcal{V}}_0 \times \tilde{\mathcal{Q}}$. We could recover the regularity requirements of the previous formulation by taking the pressure test function in $H^2(\Omega)$. However, either this choice or (6b) imply the need to construct finite element approximations to $H^2(\Omega)$, which is an involved task (at least for conforming interpolations, as we consider throughout). Thus, the finite element discretization of the PPE (6b) will not be studied. We could try to circumvent the regularity assumptions neglecting the diffusive term in (6b). Unfortunately, the simplified Poisson equation is ill-posed, as commented above.

2.3. Discrete problem

In this section, we study the discretization of (1). For the time discretization we propose the trapezoidal rule and backward differencing schemes (BDF). For the spatial approximation we will use in this section the standard Galerkin finite element method. However, no reference will be made to the need to satisfy any compatibility condition between the velocity and pressure finite element spaces. The introduction of stabilization techniques is deferred until Section 5, where a stabilization

technique designed to allow equal velocity–pressure interpolation and convection-dominated flows will be presented.

2.3.1. Monolithic time discretization. Let us introduce some notations that we will use throughout the work. Consider a uniform partition of the time interval of size δt , and let us denote by f^n the approximation of a time-dependent function f at time level $t^n = n\delta t$. For a parameter $\theta \in [0, 1]$, we will denote

$$f^{n+\theta} = \theta f^{n+1} + (1 - \theta) f^n$$

$$\delta f^{n+1} \equiv \delta^{(1)} f^{n+1} = f^{n+1} - f^n$$

$$\delta^{(i+1)} f^{n+1} = \delta^{(i)} f^{n+1} - \delta^{(i)} f^n, \quad i = 1, 2, 3, \dots$$

The discrete operators $\delta^{(i+1)}$ are centred. We will also use the backward difference operators

$$D_k f^{n+1} = \frac{1}{\gamma_k} \left(f^{n+1} - \sum_{i=0}^{k-1} \alpha_i^k f^{n-i} \right)$$

where γ_k and α_i^k for $i = 0, \dots, k-1$ are scalar values that define the time discretization. For a k th order BDF time discretization (with $k = 1-3$) we have

$$D_1 f^{n+1} = \delta f^{n+1} = f^{n+1} - f^n$$

$$D_2 f^{n+1} = \frac{3}{2} f^{n+1} - \frac{4}{3} f^n + \frac{1}{3} f^{n-1}$$

$$D_3 f^{n+1} = \frac{11}{6} f^{n+1} - \frac{18}{11} f^n + \frac{9}{11} f^{n-1} - \frac{2}{11} f^{n-2}$$

We will also use the backward extrapolation operators

$$\tilde{f}_0^{n+1} = 0$$

$$\tilde{f}_1^{n+1} = f^n$$

$$\tilde{f}_2^{n+1} = 2f^n - f^{n-1}$$

or, in general,

$$\tilde{f}_i^{n+1} = f^{n+1} - \delta^{(i)} f^{n+1} = f^{n+1} + \mathcal{O}(\delta t^i), \quad i = 0, 1, 2, \dots \quad (7)$$

For the time integration of problem (5) we consider two types of finite difference approximations. The first is the generalized trapezoidal rule, which consists of solving the following problem:

from known \mathbf{u}^n , find $\mathbf{u}^{n+1} \in \mathcal{V}_0$ and $p^{n+1} \in \mathcal{Q}$ such that

$$\left(\frac{1}{\delta t} D_1 \mathbf{u}^{n+1} + \mathbf{u}^{n+\theta} \cdot \nabla \mathbf{u}^{n+\theta}, \mathbf{v} \right) + \nu (\nabla \mathbf{u}^{n+\theta}, \nabla \mathbf{v}) - (p^{n+1}, \nabla \cdot \mathbf{v}) = \langle \mathbf{f}^{n+\theta}, \mathbf{v} \rangle \quad (8a)$$

$$(q, \nabla \cdot \mathbf{u}^{n+\theta}) = 0 \quad (8b)$$

for all $[\mathbf{v}, q] \in \mathcal{V}_0 \times \mathcal{Q}$. The force term $\mathbf{f}^{n+\theta}$ in (8a) and below has to be understood as the time average of the force in the interval $[t^n, t^{n+1}]$, even though we use a superscript $n+\theta$ to characterize it. The pressure value computed here has been identified as the pressure evaluated at t^{n+1} , although this is irrelevant for the velocity approximation. The values of interest of θ are $\theta = \frac{1}{2}$, corresponding to the second-order Crank–Nicolson scheme, and $\theta = 1$, which corresponds to the backward Euler method.

Backward differencing (BDF) time integration schemes will also be considered for the methods proposed later on. The first-order one (BDF1) coincides with the backward Euler method. BDF1 and the second-order scheme BDF2 are \mathcal{A} -stable. Higher-order methods do not keep this interesting property anymore, limitation known as *second Dahlquist barrier*. Nevertheless, BDF3 holds a less demanding $\mathcal{A}(\alpha)$ -stability property, with $\alpha = 86^\circ$, that makes these methods still appropriate. See [10] for a complete exposition of BDF methods and their stability properties.

For the second-order scheme BDF2, \mathbf{u}^1 can be computed from the backward Euler method, whereas for $n \geq 1$ the unknowns $\mathbf{u}^{n+1} \in \mathcal{V}_0$ and $p^{n+1} \in \mathcal{Q}$ are found by solving the problem

$$\left(\frac{1}{\delta t} D_2 \mathbf{u}^{n+1} + \mathbf{u}^{n+1} \cdot \nabla \mathbf{u}^{n+1}, \mathbf{v} \right) + \nu (\nabla \mathbf{u}^{n+1}, \nabla \mathbf{v}) - (p^{n+1}, \nabla \cdot \mathbf{v}) = \langle \mathbf{f}^{n+1}, \mathbf{v} \rangle$$

$$(q, \nabla \cdot \mathbf{u}^{n+1}) = 0$$

for all $[\mathbf{v}, q] \in \mathcal{V}_0 \times \mathcal{Q}$. The BDF3 also requires \mathbf{u}^2 for being initialized.

For a k th order BDF method, the unknowns $\mathbf{u}^{n+1} \in \mathcal{V}_0$ and $p^{n+1} \in \mathcal{Q}$, with $n+1 \geq k$, are found by solving the problem

$$\left(\frac{1}{\delta t} D_k \mathbf{u}^{n+1} + \mathbf{u}^{n+1} \cdot \nabla \mathbf{u}^{n+1}, \mathbf{v} \right) + \nu (\nabla \mathbf{u}^{n+1}, \nabla \mathbf{v}) - (p^{n+1}, \nabla \cdot \mathbf{v}) = \langle \mathbf{f}^{n+1}, \mathbf{v} \rangle \quad (9a)$$

$$(q, \nabla \cdot \mathbf{u}^{n+1}) = 0 \quad (9b)$$

for all $[\mathbf{v}, q] \in \mathcal{V}_0 \times \mathcal{Q}$.

The backward Euler method will be used for the first-order PC methods in time, whereas for second-order methods we will consider both the Crank–Nicolson and the BDF2 schemes. The BDF3 time integration scheme can be used for third-order VC methods with appropriate extrapolations in order to reach a third-order VC method. However, as we note below, these methods become unstable in some cases and have not been considered in the numerical experimentation.

2.3.2. Finite element discretization. Let Θ_h denote a finite element partition of the domain Ω of diameter h , from which we construct the finite element spaces \mathcal{Q}_h and $\mathcal{V}_{h,0}$, approximations to \mathcal{Q} and \mathcal{V}_0 , respectively. The former is made up with continuous functions and the other with continuous vector functions verifying the homogeneous Dirichlet boundary conditions. In the following, finite element functions will be identified with a subscript h .

The discrete problem is obtained by approximating \mathbf{u} and p . We assume that \mathbf{u}_h^n and p_h^n are constructed using the standard finite element interpolation from the nodal values. These are solutions of the nonlinear algebraic system

$$\mathbf{M} \frac{1}{\delta t} D_k \mathbf{U}^{n+1} + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (10a)$$

$$\mathbf{D} \mathbf{U}^{n+1} = 0 \quad (10b)$$

where \mathbf{U}^{n+1} and \mathbf{P}^{n+1} are the arrays of nodal values for \mathbf{u}_h^{n+1} and p_h^{n+1} , respectively, obtained from the monolithic system at the time step $n + 1$. Let us recall the matrices needed for the fully discrete problem. If we denote the node indexes with superscripts a, b , the space indexes with subscripts i, j , and the standard shape function of node a by N^a , the components of the arrays involved in these equations are

$$\mathbf{M}_{ij}^{ab} = (N^a, N^b) \delta_{ij} \quad (\delta_{ij} \text{ is the Kronecker } \delta)$$

$$\mathbf{K}(\mathbf{U}^{n+\theta})_{ij}^{ab} = (N^a, \mathbf{u}_h^{n+\theta} \cdot \nabla N^b) \delta_{ij} + \frac{1}{2} (N^a, (\nabla \cdot \mathbf{u}_h^{n+\theta}) N^b) \delta_{ij} + \nu (\nabla N^a, \nabla N^b) \delta_{ij}$$

$$\mathbf{G}_i^{ab} = (N^a, \partial_i N^b)$$

$$\mathbf{D}_j^{ab} = (N^a, \partial_j N^b)$$

$$\mathbf{F}_i^a = (N^a, f_i)$$

It is understood that all the arrays are matrices (except \mathbf{F} , which is a vector) whose components are defined by giving the right indexes (b and possibly j). Likewise, system (10) needs to be modified to account for the Dirichlet boundary conditions (matrix \mathbf{G} can be replaced by $-\mathbf{D}^t$ when this is done). Observe also that we have used the skew-symmetric form of the convective term, which yields the convective contribution to matrix $\mathbf{K}(\mathbf{U}^{n+\theta})$ skew symmetric.

2.3.3. A discrete pressure Poisson equation. At the continuous level, the PPE is not an appropriate equation to be discretized as it has been commented above. Here, we propose a DPPE obtained in a matrix setting from the fully discretized form of the classical incompressible Navier–Stokes equations, after multiplying (10a) by $\delta t \mathbf{D} \mathbf{M}^{-1}$ and invoking (10b) in the resulting equation. Replacing the discrete continuity equation (10b) by the DPPE, the equivalent monolithic scheme is

$$\mathbf{M} \frac{1}{\delta t} D_k \mathbf{U}^{n+1} + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (11a)$$

$$\gamma_k \delta t \mathbf{D} \mathbf{M}^{-1} \mathbf{G} \mathbf{P}^{n+1} = \gamma_k \delta t \mathbf{D} \mathbf{M}^{-1} (\mathbf{F}^{n+1} - \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1}) + \mathbf{D} \left(\sum_{i=0}^{k-1} \alpha_k^i \mathbf{U}^{n-i} \right) \quad (11b)$$

for a k th order BDF time discretization. The parameters γ_k and α_k^i for $i = 0, \dots, k-1$ that define this time discretization have been introduced in Section 2.3.1 for $k = 1-3$. Note that matrix $\mathbf{D} \mathbf{M}^{-1} \mathbf{G}$ is

a discrete version of the Laplacian operator. Thus, Equation (11b) is a discretization of the Poisson equation, although not in the most usual way.

Obviously, for the obtention of scheme (11) proposed herein, *no extra regularity conditions are required*, a main difference in comparison to the continuous PPE (2b). This scheme, which has been obtained after algebraic manipulation, *is equivalent to the original monolithic discretized scheme* (10). Furthermore, boundary conditions arise naturally from the original scheme.

2.3.4. Approximation of $DM^{-1}G$. A straightforward implementation of (11) is not affordable in most cases. Implementation aspects for this scheme are studied later. But the critical point is the approximation to the discrete Laplace operator $DM^{-1}G$, which is a dense matrix in general. It would be computationally expensive even using a diagonal matrix M . At this point, a further approximation can be introduced in order to avoid this system matrix. As it is typically done for PC methods, we can approximate

$$DM^{-1}G \approx L \quad \text{with components } L^{ab} = -(\nabla N^a, \nabla N^b) \quad (12)$$

where the matrix L is the standard approximation to the Laplace operator, only possible when continuous pressure interpolations are employed. Here, we propose an *enhanced* approximation

$$DM^{-1}GP^{n+1} = LP^{n+1} + (DM^{-1}G - L)P^{n+1} \approx LP^{n+1} + (DM^{-1}G - L)\tilde{P}_p^{n+1} \quad (13)$$

where \tilde{P}_p^{n+1} is a p th order backward extrapolation of P^{n+1} obtained from previous known values (see (7)). This new approximation couples space and time errors, yielding better accuracy. Using (13) in the DPPE we get

$$\gamma_k \delta t L(P^{n+1} - \tilde{P}_p^{n+1}) = \gamma_k \delta t DM^{-1}(F^{n+1} - K(U^{n+1})U^{n+1} - G\tilde{P}_p^{n+1}) + D \left(\sum_{i=0}^{k-1} \alpha_k^i U^{n-k} \right) \quad (14)$$

which is more appealing than (11b) from a computational point of view. This new scheme is appropriate in order to obtain pressure segregation and predictor–corrector methods.

3. VELOCITY-CORRECTION METHODS BASED ON A DPPE

3.1. Formulation of the schemes

Instead of obtaining a pressure segregation method from the monolithic system (10), we start from system (11), where the continuity equation has been replaced by the DPPE. Furthermore, we consider a method where the extrapolated variable is the velocity instead of the pressure. In a first step, the pressure is obtained from the approximated DPPE (14) using a backward extrapolation of order q of the velocity U^{n+1} , denoted by \tilde{U}_q^{n+1} . Once the pressure is computed, U^{n+1} is obtained from the momentum equation. For a k th order method using BDF, the split scheme reads as follows:

$$\gamma_k \delta t DM^{-1}GP^{n+1} = \gamma_k \delta t DM^{-1}(F^{n+1} - K(\tilde{U}_q^{n+1})\tilde{U}_q^{n+1}) + D \left(\sum_{i=0}^{k-1} \alpha_k^i U^{n-k} \right) \quad (15a)$$

$$M \frac{1}{\delta t} (D_k U^{n+1}) + K(U^{n+1})U^{n+1} + GP^{n+1} = F^{n+1} \quad (15b)$$

with $q = k - 1$. Invoking approximation (13) for $\mathbf{DM}^{-1}\mathbf{G}$ we obtain the following system:

$$\gamma_k \delta t \mathbf{L}(\mathbf{P}^{n+1} - \tilde{\mathbf{P}}_p^{n+1}) = \gamma_k \delta t \mathbf{DM}^{-1}(\mathbf{F}^{n+1} - \mathbf{K}(\tilde{\mathbf{U}}_q^{n+1})\tilde{\mathbf{U}}_q^{n+1} - \mathbf{G}\tilde{\mathbf{P}}_p^{n+1}) + \mathbf{D} \left(\sum_{i=0}^{k-1} \alpha_k^i \mathbf{U}^{n-i} \right) \quad (16a)$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{D}_k \mathbf{U}^{n+1}) + \mathbf{K}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + \mathbf{G}\mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (16b)$$

with $q = p = k - 1$. For instance, we can obtain a first-order method (in time) taking $k = 1$ and $q = p = 0$

$$\delta t \mathbf{LP}^{n+1} = \delta t \mathbf{DM}^{-1} \mathbf{F}^{n+1} + \mathbf{DU}^n \quad (17a)$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + \mathbf{G}\mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (17b)$$

We can obtain second-order accuracy in time with $k = 2$ and $q = p = 1$. In this case, the system to be solved is

$$\frac{2}{3} \delta t \mathbf{L}(\mathbf{P}^{n+1} - \mathbf{P}^n) = \frac{2}{3} \delta t \mathbf{DM}^{-1}(\mathbf{F}^{n+1} - \mathbf{K}(\mathbf{U}^n)\mathbf{U}^n - \mathbf{G}\mathbf{P}^n) + \mathbf{D} \left(\frac{4}{3} \mathbf{U}^n - \frac{1}{3} \mathbf{U}^{n-1} \right) \quad (18a)$$

$$\mathbf{M} \frac{1}{2\delta t} (3\mathbf{U}^{n+1} - 4\mathbf{U}^n + \mathbf{U}^{n-1}) + \mathbf{K}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + \mathbf{G}\mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (18b)$$

Remark 1

An interesting feature of these methods is that the splitting error is related to the accuracy of the velocity instead of the pressure (as it happens for PC methods). This is an advantage because it is known from the convergence analysis of different pressure segregation methods that the error estimates for the velocity are sharper than those for the pressure.

Similarly, third-order methods can be obtained with $k = 3$ (that is a BDF3 time integration scheme, and $q = p = 2$). Unfortunately, numerical experimentation dictates that this method is only conditionally stable, as it happens for third-order PC methods.

Remark 2

Numerical experimentation dictates that VC methods are *unconditionally stable* for $q \leq 1$ (this is proved for some schemes in Section 7). The instabilities shown by higher-order schemes seem to be a common feature for different sorts of methods that segregate the calculation of the velocity and pressure. This behaviour has been pointed out for PC methods in [11]. This misbehaviour is also commented in [5] for a different version of VC methods without any definitive conclusion.

3.2. Equivalent stabilized monolithic formulation

As for PC methods in [6], we could rewrite this system in a monolithic format in order to identify the perturbation terms introduced by the splitting. Taking the difference between (16b) after being multiplied by $\delta t \mathbf{DM}^{-1}$ and (16a), we get the following equivalent system:

$$\mathbf{M} \frac{1}{\delta t} \mathbf{D}_k \mathbf{U}^{n+1} + \mathbf{K}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} + \mathbf{G}\mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (19a)$$

$$\mathbf{DU}^{n+1} + \beta \delta t \mathbf{B}(\mathbf{P}^{n+1} - \tilde{\mathbf{P}}_p^{n+1}) + \delta t \mathbf{DM}^{-1}(\mathbf{K}(\mathbf{U}^{n+1})\mathbf{U}^{n+1} - \mathbf{K}(\tilde{\mathbf{U}}_q^{n+1})\tilde{\mathbf{U}}_q^{n+1}) = 0 \quad (19b)$$

where $B := DM^{-1}G - L$ and $\beta = 1$ if approximation (13) is used and 0 otherwise. The perturbed system obtained in this case is different to the one obtained for PC methods (see [6]), the main advantage being that the momentum equation is not perturbed anymore. This splitting is only introducing a perturbation in the continuity equation. The term $\delta t B(P^{n+1} - \tilde{P}_p^{n+1})$ (that also appears in the classical PC scheme with approximation (12)) arises from approximation (13) and is not a splitting error.

Remark 3

The only term due to the splitting is $\delta t DM^{-1}(K(U^{n+1})U^{n+1} - K(\tilde{U}_q^{n+1})\tilde{U}_q^{n+1})$, which is formally of order $\mathcal{O}(\delta t^{q+1})$.

4. PREDICTOR–CORRECTOR METHODS BASED ON A DPPE

4.1. Schemes with a single iterative loop

Straightforward from the monolithic scheme (11), where the mass conservation is imposed by the DPPE, we can design new predictor–corrector methods. Denoting by a superscript i the i th iteration of the scheme, the resulting predictor–corrector method is, for $k = 1$ and using a Picard linearization of the convective term

$$\delta t DM^{-1}GP^{n+1,i+1} = \delta t DM^{-1}(F^{n+1} - K(U^{n+1,i})U^{n+1,i}) + D(U^n) \quad (20a)$$

$$M \frac{1}{\delta t} (U^{n+1,i+1} - U^n) + K(U^{n+1,i})U^{n+1,i+1} + GP^{n+1,i+1} = F^{n+1} \quad (20b)$$

or, when using approximation (13),

$$\delta t L(P^{n+1,i+1} - P^{n+1,i}) = \delta t DM^{-1}(F^{n+1} - K(U^{n+1,i})U^{n+1,i} - GP^{n+1,i}) + D(U^n) \quad (21a)$$

$$M \frac{1}{\delta t} (U^{n+1,i+1} - U^n) + K(U^{n+1,i})U^{n+1,i+1} + GP^{n+1,i+1} = F^{n+1} \quad (21b)$$

Register for free at <https://www.scipedia.com> to download the version without the watermark

These methods have to be properly initialized, that is, to say, we have to start the process with a splitting error at least of the same order as the scheme. Thus, for the first-order methods we could take $U^{n+1,0} = \tilde{U}_q^{n+1}$ and $P^{n+1,0} = \tilde{P}_p^{n+1}$, with $q = p = 0$. However, it is better to use $q = p = 1$, starting with a second-order splitting error. This does not imply a significant additional computational cost and improves the initial guess of the iterative process.

In this method, all the terms are motivated from the monolithic version of the problem, a difference with some predictor–corrector methods based on PC schemes (see [7]). Thus, from the DPPE version of the monolithic problem a predictor–corrector scheme naturally arises, while it does not occur so for the typical monolithic version.

The second-order method using BDF2 is

$$\begin{aligned} \frac{2}{3}\delta t L(P^{n+1,i+1} - P^{n+1,i}) &= \frac{2}{3}\delta t DM^{-1}(F^{n+1} - K(U^{n+1,i})(U^{n+1,i}) - GP^{n+1,i}) \\ &\quad + D(\frac{4}{3}U^n - \frac{1}{3}U^{n-1}) \end{aligned} \quad (22a)$$

$$M \frac{1}{2\delta t} (3U^{n+1,i+1} - 4U^n + U^{n-1}) + K(U^{n+1,i})U^{n+1,i+1} + GP^{n+1,i+1} = F^{n+1} \quad (22b)$$

with the appropriate initializations $U^{n+1,0} = \tilde{U}_q^{n+1}$ and $P^{n+1,0} = \tilde{P}_p^{n+1}$, with $q = p = 1$.

In these schemes velocity and pressure computations are uncoupled. The iterative loop to couple velocity and pressure has been used also for a Picard linearization of the convective term in the momentum equation, although there is the possibility to use nested loops. This alternative is studied later.

As for PC schemes, we can analogously obtain the perturbed monolithic version of the predictor–corrector schemes (20) and (21). We have

$$M \frac{1}{\delta t} (U^{n+1,i+1} - U^n) + K(U^{n+1,i+1}) U^{n+1,i+1} + G P^{n+1} = F^{n+1} \quad (23a)$$

$$DU^{n+1,i+1} + \delta t \beta B(P^{n+1,i+1} - P^{n+1,i}) + \delta t D M^{-1} (K(U^{n+1,i}) U^{n+1,i+1} - K(U^{n+1,i}) U^{n+1,i}) = 0 \quad (23b)$$

where $\beta = 1$ when using approximation (13) and $\beta = 0$ otherwise. The perturbation terms disappear as the iterative procedure converges, tending to the solution of the monolithic system (10).

In [12] and [13] PC methods and their predictor–corrector counterparts have been understood as *incomplete block LU factorizations* of the monolithic matrix. Similarly, the previous predictor–corrector method can be interpreted as *a Block Gauss–Seidel iterative solver applied to the monolithic matrix*.

A third-order predictor–corrector method could also be considered using BDF3 as time integration scheme. Furthermore, in order to start with a third-order splitting error, the methods have to be initialized with second-order extrapolations ($q = p = 2$). In this case, the instability of the third-order VC method is overcome by the iterative process.

The convergence of these methods towards the monolithic system is verified in Section 8 using numerical experimentation.

4.2. Schemes with nested iterative loops

Instead of treating the nonlinearity with the same loop as for the coupling of variables, we could design a scheme using nested loops, an outer loop for the coupling and an inner loop for the nonlinearity. For the BDF1 time integration scheme with approximation (13), this new system could be written as

$$\delta t L(P^{n+1,i+1} - P^{n+1,i}) = \delta t D M^{-1} (F^{n+1} - K(U^{n+1,i}) U^{n+1,i} - G P^{n+1,i}) + D(U^n) \quad (24a)$$

$$M \frac{1}{\delta t} (U^{n+1,i+1} - U^n) + K(U^{n+1,i+1}) U^{n+1,i+1} + G P^{n+1,i+1} = F^{n+1} \quad (24b)$$

where the superscript i denotes the i th iteration of the outer loop. The momentum equation has not been linearized yet.

Similarly, using BDF2 for the time integration we would have the scheme

$$\begin{aligned} \frac{2}{3} \delta t L(P^{n+1,i+1} - P^{n+1,i}) &= \frac{2}{3} \delta t D M^{-1} (F^{n+1} - K(U^{n+1,i}) U^{n+1,i} - G P^{n+1,i}) \\ &\quad + D(\frac{4}{3} U^n - \frac{1}{3} U^{n-1}) \end{aligned} \quad (25a)$$

$$\frac{1}{2 \delta t} (3 U^{n+1,i+1} - 4 U^n + U^{n-1}) + K(U^{n+1,i+1}) U^{n+1,i+1} + G P^{n+1,i+1} = F^{n+1} \quad (25b)$$

These predictor–corrector schemes could seem more expensive than the methods introduced in the previous section. However, a faster convergence of the predictor–corrector loop could make them interesting.

Identifying $\mathbf{P}^{n+1,i}$ as $\tilde{\mathbf{P}}_p^{n+1}$, and $\mathbf{U}^{n+1,i}$ as $\tilde{\mathbf{U}}_q^{n+1}$, the system to be solved at each iteration is equivalent to the split method (18) proposed here taking as approximation for the unknown variables the solution of the previous outer iteration.

5. STABILIZED VELOCITY-CORRECTION METHODS

In this section, we consider the stabilization of the previous methods using the orthogonal subgrid scale (OSS) finite element method described in detail in [8].

Let us formulate the OSS formulation for problem (9a)–(9b). The idea is to add a least-squares form of the component of the convective and pressure gradient terms orthogonal to the velocity finite element space without boundary conditions. Let P_h be the L^2 projection onto this space. The term to be added to the discrete variational form of the problem is

$$\begin{aligned} & (P_h^\perp(\mathbf{u}_h^{n+1} \cdot \nabla \mathbf{v}_h^{n+1} + \nabla q_h^{n+1}), \tau P_h^\perp(\mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1} + \nabla p_h^{n+1})) \\ &= (\mathbf{u}_h^{n+1} \cdot \nabla \mathbf{v}_h^{n+1} + \nabla q_h^{n+1}, \tau P_h^\perp(\mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1} + \nabla p_h^{n+1})) \end{aligned} \quad (26)$$

where τ is the stabilization parameter on which the formulation depends, that we compute as

$$\tau = \left(c_1 \frac{v}{h^2} + c_2 \frac{|\mathbf{u}_h^{n+1}|}{h} \right)^{-1} \quad (27)$$

where c_1 and c_2 are algorithmic constants. Both in (26) and (27) the advection velocity \mathbf{u}_h^{n+1} is assumed to be known. It can be taken as a value computed in a previous iteration within an iterative loop or as an extrapolation from velocity values at previous time steps. Note that in (27) the Euclidean norm of this velocity appears, and therefore, τ needs to be computed at each integration point.

Adding (26) to the discrete counterpart of (9a)–(9b) leads to the following stabilized problem: find $\mathbf{u}_h^{n+1} \in \mathcal{V}_{h,0}$ and $p_h^{n+1} \in \mathcal{Q}_h$ such that

$$\begin{aligned} & \left(\frac{1}{\delta t} D_k \mathbf{u}_h^{n+1} + \mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1}, \mathbf{v}_h \right) + v(\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h) - (p_h^{n+1}, \nabla \cdot \mathbf{v}_h) \\ &+ (\mathbf{u}_h^{n+1} \cdot \nabla \mathbf{v}_h^{n+1}, \tau P_h^\perp(\mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1} + \nabla p_h^{n+1})) = \langle \mathbf{f}^{n+1}, \mathbf{v}_h \rangle \end{aligned} \quad (28a)$$

$$(q_h, \nabla \cdot \mathbf{u}_h^{n+1}) + (\nabla q_h^{n+1}, \tau P_h^\perp(\mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1} + \nabla p_h^{n+1})) = 0 \quad (28b)$$

for all $[\mathbf{v}_h, q_h] \in \mathcal{V}_{h,0} \times \mathcal{Q}_h$. Our objective in the remaining of this section is *to extend the previous VC methods to include the OSS stabilization* we have just described.

5.1. Matrix version of the stabilized monolithic system

We start writing the orthogonal projection of the convective and pressure gradient terms as

$$P_h^\perp(\mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1} + \nabla p_h^{n+1}) = \mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1} + \nabla p_h^{n+1} - \mathbf{y}_h^{n+1}$$

where \mathbf{y}_h^{n+1} is the projection of these terms onto the finite element space, that is,

$$(\mathbf{y}_h^{n+1}, \mathbf{v}_h) = (\mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1} + \nabla p_h^{n+1}, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathcal{V}_h \quad (29)$$

From these expressions it is easily checked that the discrete variational problem (28) together with the projection equation (29) lead to the nonlinear algebraic system

$$\begin{aligned} \mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{G} \mathbf{P}^{n+1} \\ + \mathbf{S}_{uu}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{S}_{up}(\mathbf{U}^{n+1}) \mathbf{P}^{n+1} - \mathbf{S}_{uy}(\mathbf{U}^{n+1}) \mathbf{Y}^{n+1} = \mathbf{F}^{n+1} \end{aligned} \quad (30a)$$

$$\mathbf{D} \mathbf{U}^{n+1} + \mathbf{S}_{pp} \mathbf{P}^{n+1} + \mathbf{S}_{pu}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} - \mathbf{S}_{py} \mathbf{Y}^{n+1} = \mathbf{0} \quad (30b)$$

$$\mathbf{M} \mathbf{Y}^{n+1} - \mathbf{C}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} - \mathbf{G} \mathbf{P}^{n+1} = \mathbf{0} \quad (30c)$$

where \mathbf{Y} is an array with the unknown nodal values of \mathbf{y} . We have adopted a BDF1 scheme in time in order to simplify the exposition. Extension to other time integration schemes is straightforward. In the notation used above, we have introduced the new stabilization matrices:

$$\mathbf{S}_{uu}(\mathbf{U}^{n+1})_{ij}^{ab} = (\tau \mathbf{u}_h^{n+1} \cdot \nabla N^a, \mathbf{u}_h^{n+1} \cdot \nabla N^b) \delta_{ij}$$

$$\mathbf{S}_{up}(\mathbf{U}^{n+1})_i^{ab} = (\tau \mathbf{u}_h^{n+1} \cdot \nabla N^a, \partial_i N^b)$$

$$\mathbf{S}_{uy}(\mathbf{U}^{n+1})_{ij}^{ab} = (\tau \mathbf{u}_h^{n+1} \cdot \nabla N^a, N^b) \delta_{ij}$$

$$\mathbf{S}_{pp}^{ab} = (\tau \nabla N^a, \nabla N^b)$$

$$\mathbf{S}_{pu}(\mathbf{U}^{n+1})_j^{ab} = (\tau \partial_j N^a, \mathbf{u}_h^{n+1} \cdot \nabla N^b)$$

$$\mathbf{S}_{py}_j^{ab} = (\tau \partial_j N^a, N^b)$$

$$\mathbf{C}(\mathbf{U}^{n+1})_{ij}^{ab} = (N^a, \mathbf{u}_h^{n+1} \cdot \nabla N^b) \delta_{ij}$$

Stabilized PC and predictor–corrector systems from system (30) have been obtained in [7]. The *enhanced* stability properties of these methods have been fully discussed in [6].

5.2. Stabilized velocity-correction system

Again, we can obtain a DPPE from the stabilized monolithic system (30). Instead of (11b) what we now obtain, for $k = 1$, is

$$\begin{aligned} (\delta t \mathbf{D} \mathbf{M}^{-1} \mathbf{G} - \mathbf{S}_{pp}) \mathbf{P}^{n+1} &= \delta t \mathbf{D} \mathbf{M}^{-1} (\mathbf{F}^{n+1} - \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} - \mathbf{S}_{uu}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} \\ &\quad - \mathbf{S}_{up}(\mathbf{U}^{n+1}) \mathbf{P}^{n+1} + \mathbf{S}_{uy}(\mathbf{U}^{n+1}) \mathbf{Y}^{n+1}) + \mathbf{D} \mathbf{U}^n \\ &\quad + \mathbf{S}_{pu}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} - \mathbf{S}_{py} \mathbf{Y}^{n+1} \end{aligned} \quad (31)$$

We need to make some further approximations in order to obtain a computationally appealing stabilized VC system. As in Section 3, the velocity in the right-hand side of (31) is extrapolated. Now, this extrapolation is needed not only for the viscous and convective term, but also for the stabilization terms associated with the momentum and continuity equations. Further, we need to extrapolate the projection array \mathbf{Y} . It has been shown in [14, 15] that by treating the projection term explicitly for the monolithic system even better stability results are obtained. In Section 8, the orthogonal projection has been treated explicitly for VC methods. For VC predictor–corrector schemes, this projection is treated implicitly due to the fact that its computation is inside the external loop. In the worst case, the error induced by these extrapolations is of order $\mathcal{O}(\tau \delta t)$. If we assume that $\tau \leq C \delta t$, we do not spoil the accuracy for first- and second-order methods.

For the momentum equation no assumptions are required. Thus, the stabilized version of system (15) is

$$\begin{aligned} (\delta t \mathbf{D} \mathbf{M}^{-1} \mathbf{G} - \mathbf{S}_{pp}) \mathbf{P}^{n+1} &= \delta t \mathbf{D} \mathbf{M}^{-1} (\mathbf{F}^{n+1} - \mathbf{K}(\tilde{\mathbf{U}}_q^{n+1}) \tilde{\mathbf{U}}_q^{n+1} - \mathbf{S}_{uu}(\tilde{\mathbf{U}}_q^{n+1}) \tilde{\mathbf{U}}_q^{n+1} \\ &\quad - \mathbf{S}_{up}(\tilde{\mathbf{U}}_q^{n+1}) \tilde{\mathbf{P}}_p^{n+1} + \mathbf{S}_{uy}(\tilde{\mathbf{U}}_q^{n+1}) \mathbf{Y}^n) + \mathbf{D} \mathbf{U}^n \\ &\quad + \mathbf{S}_{pu}(\tilde{\mathbf{U}}_q^{n+1}) \tilde{\mathbf{U}}_q^{n+1} - \mathbf{S}_{py} \mathbf{Y}^n \end{aligned} \quad (32a)$$

$$\begin{aligned} \mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{G} \mathbf{P}^{n+1} \\ + \mathbf{S}_{uu}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{S}_{up}(\mathbf{U}^{n+1}) \mathbf{P}^{n+1} - \mathbf{S}_{uy}(\mathbf{U}^{n+1}) \mathbf{Y}^n = \mathbf{F}^{n+1} \end{aligned} \quad (32b)$$

$$\mathbf{M} \mathbf{Y}^{n+1} - \mathbf{C}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} - \mathbf{G} \mathbf{P}^{n+1} = \mathbf{0} \quad (32c)$$

where we have adopted a BDF1 time integration scheme. At this point, approximation (13) can be applied in order to avoid dealing with $\mathbf{D} \mathbf{M}^{-1} \mathbf{G}$.

5.3. Stabilized predictor–corrector scheme

As shown in Section 4, the monolithic system solution can be obtained using a predictor–corrector method, interpreted as a block Gauss–Seidel iterative procedure. Using similar arguments to those used for the VC method presented above, we straightforwardly get the stabilized version of the predictor–corrector system (20) by iterating over (32), with inner or outer iteration loops (as in Section 4).

6. IMPLEMENTATION ASPECTS

At this point, we discuss how to treat non-homogeneous boundary conditions when using VC methods. For the sake of clarity we use a BDF1 time integration scheme and neglect the stabilization terms.

Let us introduce some new notation. Given an array $\tilde{\mathbf{X}}$ with all the nodal values of an unknown \mathbf{x} , we split this array in two parts, \mathbf{X} associated with free nodes and \mathbf{X}_d associated with the nodes belonging to the Dirichlet boundary for the velocity, that is,

$$\tilde{\mathbf{X}} = [\mathbf{X}, \mathbf{X}_d]^T$$

Furthermore, for every matrix $\tilde{\mathbf{A}} = [\tilde{\mathbf{A}}^{ab}]$, a and b being nodes of the mesh, we define the block $\mathbf{A} = [\mathbf{A}^{ab}]$, where a and b are free nodes, and $\mathbf{A}_d = [\mathbf{A}_d^{ab}]$, where a is a free node and b a fixed node. In all cases, fixed or free is understood with respect to the velocity unknown.

The discrete momentum equation for non-homogeneous Dirichlet boundary conditions reads

$$\begin{aligned} \mathbf{M} \frac{1}{\delta t} \mathbf{U}^{n+1} + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{G} \mathbf{P}^{n+1} \\ = \mathbf{F}^{n+1} - \mathbf{M}_d \frac{1}{\delta t} (\mathbf{U}_d^{n+1} - \mathbf{U}_d^n) - \mathbf{K}_d(\mathbf{U}^{n+1}) \mathbf{U}_d^{n+1} + \mathbf{M} \frac{1}{\delta t} \mathbf{U}^n \\ =: \mathbf{F}_+^{n+1} \end{aligned} \quad (33)$$

where \mathbf{F}_+ accounts for the force terms together with Dirichlet boundary conditions and time integration terms.

On the other hand, with the notation introduced above, we can rewrite the discrete continuity equation as

$$\mathbf{D} \mathbf{U}^{n+1} = -\mathbf{D}_d \mathbf{U}_d^{n+1} \quad (34)$$

Multiplying (33) by $\delta t \mathbf{M}^{-1} \mathbf{G}$ and invoking (34) we get

$$\delta t \mathbf{D} \mathbf{M}^{-1} \mathbf{G} \mathbf{P}^{n+1} = \mathbf{D} \mathbf{M}^{-1} (\mathbf{F}_+^{n+1} - \mathbf{K}(\tilde{\mathbf{U}}_q^{n+1}) \tilde{\mathbf{U}}_q^{n+1}) + \mathbf{D}_d \mathbf{U}_d^{n+1} + \mathbf{D} \mathbf{U}^n \quad (35)$$

where we have extrapolated \mathbf{U}^{n+1} in order to obtain the system to be solved for the VC system. At this point, we are able to use approximation (13).

In order to solve system (33)–(35), we introduce an auxiliary nodal array \mathbf{X}^{n+1} , obtained from the following system:

$$\mathbf{M} \mathbf{X}^{n+1} = \mathbf{F}_+^{n+1} - \mathbf{K}(\tilde{\mathbf{U}}_q^{n+1}) \tilde{\mathbf{U}}_q^{n+1}$$

This array allows us to write equation (35) as

$$\delta t \mathbf{D} \mathbf{M}^{-1} \mathbf{G} \mathbf{P}^{n+1} = \mathbf{D} \mathbf{X}^{n+1} + \mathbf{D}_d \mathbf{U}_d^{n+1} + \mathbf{D} \mathbf{U}^n$$

We summarize the implementation of this VC method in a compact manner in Box 1, both with approximation (13) ($\beta = 1$) and without this approximation ($\beta = 0$). A simple fixed point

Box 1. Velocity-correction algorithm.

-
- Solve $\mathbf{M}\mathbf{X}^{n+1} = \mathbf{F}_+^{n+1} - \mathbf{K}(\tilde{\mathbf{U}}_q^{n+1})\tilde{\mathbf{U}}_q^{n+1} - \beta\mathbf{G}\tilde{\mathbf{P}}_p^{n+1}$.
 - Add $\tilde{\mathbf{X}}^{n+1} = [\mathbf{X}^{n+1}; \mathbf{0}]^T + [\mathbf{U}^n; \mathbf{U}_d^{n+1}]^T$.
- IF $\beta = 0$:
- Solve $\delta t \mathbf{D}\mathbf{M}^{-1}\mathbf{G}\mathbf{P}^{n+1} = \tilde{\mathbf{D}}\tilde{\mathbf{X}}^{n+1}$.
- ELSE:
- Solve $\delta t \mathbf{L}\mathbf{P}^{n+1} = \tilde{\mathbf{D}}\tilde{\mathbf{X}}^{n+1} + \delta t \mathbf{L}\tilde{\mathbf{P}}_p^{n+1}$.
- END IF
- Set $\mathbf{U}^{n+1,0} = \mathbf{U}^n$ and $i = 0$.
- DO UNTIL CONVERGENCE:
- Solve $[\frac{1}{\delta t}\mathbf{M} + \mathbf{K}(\mathbf{U}^{n+1,i})]\mathbf{U}^{n+1,i+1} = \mathbf{F}_+^{n+1} - \mathbf{G}\mathbf{P}^{n+1}$.
 - Set $i \leftarrow i + 1$.
- END DO

method has been used to linearize the convective term. The implementation of the rest of the predictor–corrector methods presented is straightforward, as well as the implementation of their stabilized versions.

Remark 4

We end this section with a comment about the system matrix $\mathbf{D}\mathbf{M}^{-1}\mathbf{G}$. This matrix is dense when a consistent matrix \mathbf{M} is used. The obtention of this matrix is only computationally feasible when using a diagonal matrix \mathbf{M} . However, for iterative solvers where the system matrix is not explicitly required and only matrix–vector multiplications are needed, $\mathbf{D}\mathbf{M}^{-1}\mathbf{G}$ can be treated without any extra approximation. Each matrix–vector product implies to solve a linear system with matrix \mathbf{M} .

7. STABILITY ANALYSIS OF VELOCITY-CORRECTION METHODS

In this section, we present a complete set of stability results for the original DPPE VC schemes proposed in this paper. The key aspects of the analysis are:

- To identify the inherent stability related to VC methods.
- The fact that stability properties are weaker for more accurate schemes.

We analyse VC methods using BDF1 and Crank–Nicolson time integration schemes. For simplicity in the exposition we will identify the methods as BDF1- \mathbf{U}_q - \mathbf{P}_p , q being the order of the velocity extrapolation and p the order of the pressure extrapolation. Similarly, for the Crank–Nicolson

scheme we will use CN-U_q-P_p. The parameters q and p will take the values 0 and 1 for BDF1 and $p = q = 1$ for Crank–Nicolson.

Let us start introducing some additional notation. If \mathbf{X}, \mathbf{Y} are arrays, $\{\mathbf{X}^n\}_{n=0,1,\dots,N}$ is a sequence of arrays of $N + 1$ terms and \mathbf{A} a symmetric positive semi-definite matrix, we define

$$(\mathbf{X}, \mathbf{Y})_{\mathbf{A}} := \mathbf{X} \cdot \mathbf{A} \mathbf{Y}$$

$$\|\mathbf{x}\|_{\mathbf{A}} := (\mathbf{x} \cdot \mathbf{A} \mathbf{x})^{1/2}$$

$$\|\mathbf{Y}\|_{-\mathbf{A}} := \sup_{\mathbf{x} \neq 0} \frac{\mathbf{Y} \cdot \mathbf{x}}{\|\mathbf{x}\|_{\mathbf{A}}} \quad (\text{here } \mathbf{A} \text{ is assumed to be positive definite})$$

$$\{\mathbf{X}^n\} \in \ell^\infty(\mathbf{A}) \iff \|\mathbf{X}^n\|_{\mathbf{A}} \leq C < \infty \quad \forall n = 0, 1, \dots, N$$

$$\{\mathbf{X}^n\} \in \ell^p(\mathbf{A}) \iff \sum_{n=0}^N \delta t \|\mathbf{X}^n\|_{\mathbf{A}}^p \leq C < \infty, \quad 1 \leq p < \infty$$

Here and in the following, C denotes a positive constant, not necessarily the same at different appearances. Moreover, we denote by $N = \lceil T/\delta t \rceil$.

A remark is needed when $\mathbf{A} = \mathbf{K}$. This matrix is not symmetric, but it has the contribution from the convective term, which is skew symmetric, and the contribution from the viscous term, \mathbf{K}_{visc} , which is symmetric and positive definite. We will simply write $\mathbf{U} \cdot \mathbf{K}(\mathbf{U})\mathbf{U} = \mathbf{U} \cdot \mathbf{K}_{\text{visc}}\mathbf{U} \equiv \|\mathbf{U}\|_{\mathbf{K}}^2$.

These definitions will allow us to express our stability results in a compact manner.

For obtaining stability bounds in this section the basic assumption in all the cases will be that

$$\sum_{n=0}^N \delta t \|\mathbf{F}^n\|_{\mathbf{M}}^2 \leq C < \infty \tag{36}$$

for all $\delta t > 0$, that corresponds to requiring $\mathbf{f} \in L^2(0, T; \mathbf{L}^2(\Omega))$ for the continuous problem. Apart from this, *no other regularity assumptions will be required*.

The main stability results for the different VC methods analysed in this section have been highlighted in boxes.

7.1. Stability of methods using BDF1

The first scheme to be studied is the simplest BDF1-U0-P0, together with the scheme BDF1-U0 *without pressure extrapolation*, that is to say, without making use of approximation (13). As before, we will distinguish both possibilities according to the parameter β , and we will denote the resulting methods, by BDF1-U0-(P0). For these methods, we will obtain the following stability results:

Stability of BDF1-U0-(P0):

$$\{\mathbf{U}^n\} \in \ell^\infty(\mathbf{M}) \cap \ell^2(\mathbf{K}), \quad \{\sqrt{\delta t} \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n\} \in \ell^2(\mathbf{M}), \quad \{\sqrt{\delta t} \mathbf{P}^n\} \in \ell^2(\beta \mathbf{B})$$

Recall that matrix B is defined as $B = DM^{-1}G - L$. This estimate is optimal for the velocity. The stability for the pressure is certainly not optimal, but the important point is that we will obtain it without relying on the classical inf-sup condition for the velocity-pressure interpolation. Observe also that this stability is lost when $\beta = 0$.

In the previous estimate we have also displayed the additional control we have on the norm of the viscous plus convective terms.

First, let us write the scheme to be analysed as

$$\delta t[\beta L + (1 - \beta)DM^{-1}G]P^{n+1} = \delta tDM^{-1}F^{n+1} + DU^n \quad (37a)$$

$$M\frac{1}{\delta t}(U^{n+1} - U^n) + K(U^{n+1})U^{n+1} + GP^{n+1} = F^{n+1} \quad (37b)$$

where, as before, $\beta = 1$ using approximation (13) and $\beta = 0$ otherwise. Instead of analysing (37) we will work with an equivalent form of this method, introducing an intermediate velocity, as for PC methods. In this new format we have

$$M\frac{1}{\delta t}(\tilde{U}^{n+1} - U^n) + GP^{n+1} = F^{n+1} \quad (38a)$$

$$D\tilde{U}^{n+1} + \delta t\beta BP^{n+1} = 0 \quad (38b)$$

$$M\frac{1}{\delta t}(U^{n+1} - \tilde{U}^{n+1}) + K(U^{n+1})U^{n+1} = 0 \quad (38c)$$

Thus, the intermediate velocity we have introduced is not divergence-free when approximation (13) is used ($\beta = 1$). When $\beta = 0$, the second term in the left-hand side of (38b) disappears together with the stability results associated with the norm $\|\cdot\|_{\beta B}$.

The stability bounds for the BDF1-U0-P0 and BDF1-U0 methods are summarized in the next theorem.

Theorem 1

The following stability estimates hold for the BDF1-U0-(P0) method:

$$\max_{0 \leq n \leq N} \{\|U^n\|_M^2\} + \sum_{n=1}^N \delta t \{\|U^n\|_K^2 + \|\sqrt{\delta t}M^{-1}K(U^n)U^n\|_M^2 + \|\sqrt{\delta t}P^n\|_{\beta B}^2\} \leq C$$

for all $\delta t > 0$.

Proof

Taking the inner product of (38a) with $2\delta t\tilde{U}^{n+1}$ and using the identity

$$(2a, a - b) := a^2 - b^2 + (a - b)^2$$

we get

$$\|\tilde{U}^{n+1}\|_M^2 - \|U^n\|_M^2 + \|\tilde{U}^{n+1} - U^n\|_M^2 + 2\delta t\tilde{U}^{n+1} \cdot GP^{n+1} = 2\delta t\tilde{U}^{n+1} \cdot F^{n+1} \quad (39)$$

On the other hand, (38c) can be reordered in order to obtain

$$U^{n+1} + \delta tM^{-1}K(U^{n+1})U^{n+1} = \tilde{U}^{n+1} \quad (40)$$

Squaring both terms of this equation with the inner product $(\cdot, \cdot)_M$ it is found that

$$\|U^{n+1}\|_M^2 + 2\delta t \|U^{n+1}\|_K^2 + \|\delta t M^{-1}K(U^{n+1})U^{n+1}\|_M^2 = \|\tilde{U}^{n+1}\|_M^2 \quad (41)$$

Multiplying (40) by $2\delta t F^{n+1}$ we get

$$2\delta t U^{n+1} \cdot F^{n+1} - 2\delta t \tilde{U}^{n+1} \cdot F^{n+1} + 2\delta t^2 M^{-1}K(U^{n+1})U^{n+1} \cdot F^{n+1} = 0 \quad (42)$$

Adding up (39), (41) and subtracting (42) we obtain

$$\begin{aligned} & \|U^{n+1}\|_M^2 - \|U^n\|_M^2 + \|\tilde{U}^{n+1} - U^n\|_M^2 + 2\delta t \|U^{n+1}\|_K^2 + \|\delta t M^{-1}K(U^{n+1})U^{n+1}\|_M^2 \\ & + 2\delta t \tilde{U}^{n+1} \cdot GP^{n+1} \\ & = 2\delta t U^{n+1} \cdot F^{n+1} + 2\delta t^2 M^{-1}K(U^{n+1})U^{n+1} \cdot F^{n+1} \\ & \leq \delta t \|F^{n+1}\|_{-K}^2 + \delta t \|U^{n+1}\|_K^2 + 2\delta t^2 \|M^{-1}F^{n+1}\|_M^2 + \frac{1}{2} \|\delta t M^{-1}K(U^{n+1})U^{n+1}\|_M^2 \end{aligned} \quad (43)$$

From (38b) we can bound the term involving the pressure, obtaining

$$2\delta t \tilde{U}^{n+1} \cdot GP^{n+1} = -2\delta t P^{n+1} \cdot D\tilde{U}^{n+1} = 2\delta t^2 \|P^{n+1}\|_{\beta B}^2 \quad (44)$$

Replacing (40) in (39), and summing from $n=0$ to $N-1$ (or up to any arbitrary time level), we find that

$$\begin{aligned} & \|U^N\|_M^2 + \sum_{n=1}^N \|\tilde{U}^n - U^{n-1}\|_M^2 + \sum_{n=1}^N \delta t \|U^n\|_K^2 + \sum_{n=1}^N \delta t \|\sqrt{\delta t} M^{-1}K(U^n)U^n\|_M^2 \\ & + \sum_{n=1}^N \delta t \|\sqrt{\delta t} P^n\|_{\beta B}^2 \leq C \end{aligned} \quad (45)$$

This proves the theorem. \square

The stability analysis of the method BDF1-U0-P1 follows in a straightforward way. Obviously, this method only makes sense if approximation (13) is used ($\beta=1$, with the previous notation). The stability results we are going to obtain are

<p>Stability of BDF1-U0-P1:</p> $\{U^n\} \in \ell^\infty(M) \cap \ell^2(K), \quad \{\sqrt{\delta t} M^{-1}K(U^n)U^n\} \in \ell^2(M)$ $\{\delta t P^n\} \in \ell^\infty(B), \quad \{\sqrt{\delta t} \delta P^n\} \in \ell^2(B)$
--

Observe that the pressure stability now is weaker than for the BDF1-U0-P0 method, where the pressure was extrapolated only up to zero order. Control in $\ell^\infty(B)$ is obtained only for $\{\delta t P^n\}$, whereas the optimal would be $\{\sqrt{\delta t} P^n\}$ if $\delta t = \mathcal{O}(h^2)$. In general, a better approximation for the pressure implies less stability (which has to be found either from the use of stabilization methods or by invoking an inf-sup condition).

Now, the scheme reads as follows:

$$\delta t \mathbf{L}(\mathbf{P}^{n+1} - \mathbf{P}^n) = \delta t \mathbf{D} \mathbf{M}^{-1}(\mathbf{F}^{n+1} - \mathbf{G} \mathbf{P}^n) + \mathbf{D} \mathbf{U}^n \quad (46a)$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (46b)$$

As above, we will work with an equivalent form of this method, introducing an intermediate velocity:

$$\mathbf{M} \frac{1}{\delta t} (\tilde{\mathbf{U}}^{n+1} - \mathbf{U}^n) + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (47a)$$

$$\mathbf{D} \tilde{\mathbf{U}}^{n+1} + \delta t \mathbf{B}(\mathbf{P}^{n+1} - \mathbf{P}^n) = \mathbf{0} \quad (47b)$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \tilde{\mathbf{U}}^{n+1}) + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} = \mathbf{0} \quad (47c)$$

Theorem 2

The following stability estimates hold for the BDF1-U0-P1 method:

$$\max_{0 \leq n \leq N} \{ \|\mathbf{U}^n\|_{\mathbf{M}}^2 + \|\delta t \mathbf{P}^n\|_{\mathbf{B}}^2 \} + \sum_{n=1}^N \delta t \{ \|\mathbf{U}^n\|_{\mathbf{K}}^2 + \|\sqrt{\delta t} \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n\|_{\mathbf{M}}^2 + \|\sqrt{\delta t} \delta \mathbf{P}^n\|_{\mathbf{B}}^2 \} \leq C$$

for all $\delta t > 0$.

Proof

The proof of this theorem follows the same line than the one of Theorem 1. The only place where the analysis of this method differs from the previous one is in (44), which now has to be replaced by

$$2\delta t \tilde{\mathbf{U}}^{n+1} \cdot \mathbf{G} \mathbf{P}^{n+1} = -2\delta t \mathbf{P}^{n+1} \cdot \mathbf{D} \tilde{\mathbf{U}}^{n+1} = \delta t^2 (\|\mathbf{P}^{n+1}\|_{\mathbf{B}}^2 - \|\mathbf{P}^n\|_{\mathbf{B}}^2 + \|\delta \mathbf{P}^{n+1}\|_{\mathbf{B}}^2) \quad (48)$$

□

Now we will obtain stability results for the BDF1-U1-P0 method. If approximation (13) is not used, the stability results presented in the following for the pressure disappear. Thus, we will concentrate only in the case $\beta = 1$. The results are:

Stability of BDF1-U1-P0:

$$\{\mathbf{U}^n\} \in \ell^\infty(\mathbf{M}) \cap \ell^2(\mathbf{K}), \quad \{\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n\} \in \ell^\infty(\mathbf{M}), \quad \{\sqrt{\delta t} \mathbf{P}^n\} \in \ell^2(\mathbf{B})$$

This method reads as follows:

$$\delta t \mathbf{L} \mathbf{P}^{n+1} = \delta t \mathbf{D} \mathbf{M}^{-1}(\mathbf{F}^{n+1} - \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n) + \mathbf{D} \mathbf{U}^n \quad (49a)$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (49b)$$

The equivalent version that will be analysed is

$$\mathbf{M} \frac{1}{\delta t} (\tilde{\mathbf{U}}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (50a)$$

$$\mathbf{D} \tilde{\mathbf{U}}^{n+1} + \delta t \mathbf{B} \mathbf{P}^{n+1} = \mathbf{0} \quad (50b)$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \tilde{\mathbf{U}}^{n+1}) + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} - \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n = \mathbf{0} \quad (50c)$$

At the first time step the first-order method BDF1-U0-P0 is used for the initialization.

Theorem 3

The following stability estimates hold for the BDF1-U1-P0 method:

$$\max_{0 \leq n \leq N} \{ \|\mathbf{U}^n\|_{\mathbf{M}}^2 + \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n\|_{\mathbf{M}}^2 \} + \sum_{n=1}^N \delta t \{ \|\mathbf{U}^n\|_{\mathbf{K}}^2 + \|\sqrt{\delta t} \mathbf{P}^n\|_{\mathbf{B}}^2 \} \leq C$$

for all $\delta t > 0$.

Proof

After multiplying (50a) by $2\delta t \tilde{\mathbf{U}}^{n+1}$ we get

$$\begin{aligned} & \|\tilde{\mathbf{U}}^{n+1}\|_{\mathbf{M}}^2 - \|\mathbf{U}^n\|_{\mathbf{M}}^2 + \|\tilde{\mathbf{U}}^{n+1} - \mathbf{U}^n\|_{\mathbf{M}}^2 + 2\delta t \tilde{\mathbf{U}}^{n+1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n \\ & + 2\delta t \tilde{\mathbf{U}}^{n+1} \cdot \mathbf{G} \mathbf{P}^{n+1} = 2\delta t \tilde{\mathbf{U}}^{n+1} \cdot \mathbf{F}^{n+1} \end{aligned} \quad (51)$$

We can reorder (50c) in the form:

$$\mathbf{U}^{n+1} + \delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} = \tilde{\mathbf{U}}^{n+1} + \delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n \quad (52)$$

After squaring (52) with the inner product $(\cdot, \cdot)_{\mathbf{M}}$, we have

$$\begin{aligned} & \|\mathbf{U}^{n+1}\|_{\mathbf{M}}^2 + 2\delta t \|\mathbf{U}^{n+1}\|_{\mathbf{K}}^2 + \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1}\|_{\mathbf{M}}^2 \\ & = \|\tilde{\mathbf{U}}^{n+1}\|_{\mathbf{M}}^2 + 2\delta t \tilde{\mathbf{U}}^{n+1} \cdot \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n + \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n\|_{\mathbf{M}}^2 \end{aligned} \quad (53)$$

On the other hand, multiplying (52) by $2\delta t \mathbf{F}^{n+1}$, we get

$$\begin{aligned} & 2\delta t \mathbf{U}^{n+1} \cdot \mathbf{F}^{n+1} + 2\delta t^2 \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} \cdot \mathbf{F}^{n+1} \\ & - 2\delta t^2 \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n \cdot \mathbf{F}^{n+1} = 2\delta t \tilde{\mathbf{U}}^{n+1} \cdot \mathbf{F}^{n+1} \end{aligned} \quad (54)$$

Finally, the term involving the pressure in (51) is bounded using (44). From (51), (53), (54) and (44) we have

$$\begin{aligned} & \|\mathbf{U}^{n+1}\|_{\mathbf{M}}^2 - \|\mathbf{U}^n\|_{\mathbf{M}}^2 + \|\tilde{\mathbf{U}}^{n+1} - \mathbf{U}^n\|_{\mathbf{M}}^2 + 2\delta t \|\mathbf{U}^{n+1}\|_{\mathbf{K}}^2 + \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1}\|_{\mathbf{M}}^2 \\ & - \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n\|_{\mathbf{M}}^2 + 2\delta t \|\sqrt{\delta t} \mathbf{P}^{n+1}\|_{\mathbf{B}}^2 \end{aligned}$$

$$\begin{aligned}
&= 2\delta t \mathbf{U}^{n+1} \cdot \mathbf{F}^{n+1} + 2\delta t^2 \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} \cdot \mathbf{F}^{n+1} - 2\delta t^2 \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n \cdot \mathbf{F}^{n+1} \\
&\leq \delta t \|\mathbf{F}^{n+1}\|_{\mathbf{K}}^2 + \delta t \|\mathbf{U}^{n+1}\|_{\mathbf{K}}^2 + 4\alpha \delta t \|\mathbf{M}^{-1} \mathbf{F}^{n+1}\|_{\mathbf{M}}^2 + \frac{\delta t}{2\alpha} \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1}\|_{\mathbf{M}}^2 \\
&\quad + \frac{\delta t}{2\alpha} \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n\|_{\mathbf{M}}^2
\end{aligned} \tag{55}$$

being $\alpha := 2 \max\{t_0, \delta t\}$ and $t_0 > 0$ is a fixed time value. Summing up from $n = 0$ to $N - 1$ (or to an arbitrary time level), we obtain

$$\begin{aligned}
&\|\mathbf{U}^N\|_{\mathbf{M}}^2 + \sum_{n=1}^N \|\tilde{\mathbf{U}}^n - \mathbf{U}^{n-1}\|_{\mathbf{M}}^2 + \sum_{n=1}^N \delta t \|\mathbf{U}^n\|_{\mathbf{K}}^2 + \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^N) \mathbf{U}^N\|_{\mathbf{M}}^2 \\
&\quad + \sum_{n=1}^N \delta t \|\sqrt{\delta t} \mathbf{P}^n\|_{\mathbf{B}}^2 \leq C + \sum_{n=1}^N \frac{\delta t}{\alpha} \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n\|_{\mathbf{M}}^2
\end{aligned} \tag{56}$$

The proof of the stability results required is finished after using the discrete Gronwall inequality (see [16]) in (56). We stress the fact that $\delta t/\alpha < 1$. Therefore, the discrete Gronwall lemma can be applied without any restriction over the time step size. \square

The BDF1-U1-P1 scheme is originally written as

$$\delta t \mathbf{L}(\mathbf{P}^{n+1} - \mathbf{P}^n) = \delta t \mathbf{D} \mathbf{M}^{-1} (\mathbf{F}^{n+1} - \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n - \mathbf{G} \mathbf{P}^n) + \mathbf{D} \mathbf{U}^n \tag{57a}$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1} \tag{57b}$$

The stability analysis of this method differs from the previous one just in the pressure term to be bounded. Using (48) instead of (40) it is easily obtained that:

Stability of BDF1-U1-P1:

$$\begin{aligned}
&\{\mathbf{U}^n\} \in \ell^\infty(\mathbf{M}) \cap \ell^2(\mathbf{K}), \quad \{\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^n) \mathbf{U}^n\} \in \ell^\infty(\mathbf{M}) \\
&\{\delta t \mathbf{P}^n\} \in \ell^\infty(\mathbf{B}), \quad \{\sqrt{\delta t} \delta \mathbf{P}^n\} \in \ell^2(\mathbf{B})
\end{aligned}$$

7.2. Stability of methods using the Crank–Nicolson time integration

The Crank–Nicolson time integration scheme will be the last to be analysed. We only present the stability results of the CN-U1-P1 method, since in order to maintain the second-order accuracy of the Crank–Nicolson scheme the velocity and the pressure need to be extrapolated to first order. Again, we will consider that approximation (13) is used; otherwise, the pressure bounds presented next disappear. This method reads as follows:

$$\delta t \mathbf{L}(\mathbf{P}^{n+1} - \mathbf{P}^n) = \delta t \mathbf{D} \mathbf{M}^{-1} (\mathbf{F}^{n+1} - \mathbf{K}(\mathbf{U}^{n-1/2}) \mathbf{U}^{n-1/2} - \mathbf{G} \mathbf{P}^n) + \mathbf{D}(\mathbf{U}^n) \tag{58a}$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+1/2}) \mathbf{U}^{n+1/2} + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1} \tag{58b}$$

and its equivalent form to be analysed is

$$\mathbf{M} \frac{1}{\delta t} (\tilde{\mathbf{U}}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n-1/2}) \mathbf{U}^{n-1/2} + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1} \quad (59a)$$

$$\mathbf{D} \tilde{\mathbf{U}}^{n+1} + \delta t \mathbf{B} \mathbf{P}^{n+1} = 0 \quad (59b)$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \tilde{\mathbf{U}}^{n+1}) + \mathbf{K}(\mathbf{U}^{n+1/2}) \mathbf{U}^{n+1/2} - \mathbf{K}(\mathbf{U}^{n-1/2}) \mathbf{U}^{n-1/2} = 0 \quad (59c)$$

At the first time step, we adopt the first-order BDF1-U0-P0 for simplicity. It does not affect the overall second-order accuracy of the method. In the following setting this initialization is equivalent to take $\mathbf{U}^{-1/2} = 0$. The stability results we prove here are

<p>Stability of CN-U1-P1:</p> $\{\mathbf{U}^n\} \in \ell^\infty(\mathbf{M}) \cap \ell^2(\mathbf{K}), \quad \{\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^{n-1/2}) \mathbf{U}^{n-1/2}\} \in \ell^\infty(\mathbf{M})$ $\{\delta t \mathbf{P}^n\} \in \ell^\infty(\mathbf{B}), \quad \{\sqrt{\delta t} \delta \mathbf{P}^n\} \in \ell^2(\mathbf{B})$
--

These results are summarized in the following theorem.

Theorem 4

The following stability estimates hold for the CN-U1-P1 method

$$\begin{aligned} & \max_{0 \leq n \leq N} \{ \|\mathbf{U}^n\|_{\mathbf{M}}^2 + \|\delta t \mathbf{M}^{-1} \mathbf{K}(\mathbf{U}^{n-1/2}) \mathbf{U}^{n-1/2}\|_{\mathbf{M}}^2 + \|\delta t \mathbf{P}^n\|_{\mathbf{B}}^2 \} \\ & + \sum_{n=1}^N \delta t \{ \|\mathbf{U}^n\|_{\mathbf{K}}^2 + \|\sqrt{\delta t} \delta \mathbf{P}^n\|_{\mathbf{B}}^2 \} \leq C \end{aligned}$$

for all $\delta t > 0$.

Proof

The bounds of this theorem can be proved following the same steps as for the proof of the stability estimates for the BDF1-U1-P0 method and bounding the pressure term in the momentum equation as for the BDF1-U0-P1 method. \square

7.3. Additional pressure stability and pressure stabilization

We can easily see from the previous stability bounds that the inherent pressure stability of VC methods seems insufficient. We only have some pressure stability under approximation (13). And even in this case, the stability is under the norm associated with \mathbf{B} (a difference between discrete Laplacians that tends to zero with h). Thus, their behaviour is different from PC methods, which have a *stronger* inherent pressure stability [6, 9]. For a first-order splitting error, using PC methods with approximation (13) we have control over the whole pressure gradient $\|\sqrt{\delta t} \nabla p_h^{n+1}\|_0$. For VC methods, the bound for the projection of the pressure gradient onto the finite element space (weighted with $\sqrt{\delta t}$) does not appear. Nevertheless, we can recover the control over the whole

gradient by using the stability provided by the momentum equation (as shown in the following theorem). Summarizing, even though the stability bounds of the VC methods seem weaker, we also have stability over the whole pressure gradient (under approximation (13)).

Let us introduce some notation. We will need some projections of ∇p_h^{n+1} with respect to the L^2 -inner product

- π_1 : projection onto $\mathcal{V}_{h,0}$
- π_2 : projection onto $(\mathcal{V}_{h,0})^\perp \cap \mathcal{V}_h$
- π_3 : projection onto $(\mathcal{V}_h)^\perp$

and we will denote $\pi_{ij} := \pi_i + \pi_j$ and $\pi_h = \pi_{12}$.

Theorem 5

Assuming a quasi-uniform finite element partition, the following stability estimate holds for the BDF1-U0-P_p method:

$$\sum_{n=1}^N \delta t \|h \pi_1^n\|_0^2 \leq C \quad (60)$$

For the BDF1-U1-P_p and CN-U1-P_p, it holds

$$\sum_{n=1}^N \delta t \|h \pi_1^n\|_0 \leq C \quad (61)$$

being $p=0$ or 1 . These bounds hold for all $\delta t > 0$.

Proof

Let us write inequalities (38a), (50a) and (59a) in an appropriate format in order to analyse the different methods together. We use the following variational form:

$$\begin{aligned} & \frac{1}{\delta t} (\tilde{\mathbf{u}}_h^{n+1} - \mathbf{u}_h^n, \mathbf{v}_h) + \gamma v (\nabla \mathbf{u}_h^{n+\delta-1}, \nabla \mathbf{v}_h) + \gamma (\mathbf{u}_h^{n+\delta-1} \cdot \nabla \mathbf{u}_h^{n+\delta-1}, \mathbf{v}_h) \\ & + (\nabla p_h^{n+1}, \mathbf{v}_h) = \langle \mathbf{f}^{n+\delta}, \mathbf{v}_h \rangle \end{aligned} \quad (62)$$

which must hold for all test functions $\mathbf{v}_h \in \mathcal{V}_{h,0}$ and where $\delta = 1$ for the BDF1 scheme and $\delta = \frac{1}{2}$ for the CN scheme. Furthermore, $\gamma = 0$ for the BDF1-U0-P_p and $\gamma = 1$ for the rest of methods. Taking π_1^{n+1} as test function in (62) and using the standard inverse estimate valid for quasi-uniform finite element partitions (see [17])

$$\|\pi_1^n\|_1 \leq \frac{C_{\text{inv}}}{h} \|\pi_1^n\|_0 \quad (63)$$

C_{inv} being a constant uniform with respect to the mesh size, we have

$$\begin{aligned} \|\pi_1^{n+1}\|_0^2 & \leq \|\mathbf{f}^{n+\delta}\|_{-1} \frac{C_{\text{inv}}}{h} \|\pi_1^{n+1}\|_0 + \frac{1}{2\delta t} \|\tilde{\mathbf{u}}_h^{n+1} - \mathbf{u}_h^n\|_0 \|\pi_1^{n+1}\|_0 \\ & + \gamma (N_a \|\tilde{\mathbf{u}}_h^{n+\delta-1}\|_1 + N_c \|\tilde{\mathbf{u}}_h^{n+\delta-1}\|_1^2) \frac{C_{\text{inv}}}{h} \|\pi_1^{n+1}\|_0 \end{aligned} \quad (64)$$

where N_a and N_c are the norms of the viscous and convective terms, respectively. That is to say, we have used the inequalities

$$v(\nabla \mathbf{u}, \nabla \mathbf{v}) \leq N_a \|\mathbf{u}\|_1 \|\mathbf{v}\|_1$$

$$(\mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v}) \leq N_c \|\mathbf{u}\|_1^2 \|\mathbf{v}\|_1$$

valid for any functions $\mathbf{u}, \mathbf{v} \in \mathbf{H}_0^1(\Omega)$, where $\|\cdot\|_1$ is the norm of this space. Likewise, $\|\cdot\|_{-1}$ denotes the norm in $\mathbf{H}^{-1}(\Omega)$.

We can easily get the bounds stated in the theorem from inequality (64) bounding the right-hand side terms using the stability bounds obtained in Theorems 1–4. \square

Let us conclude by showing which is the additional stability introduced by pressure stabilization only (not by convection stabilization). This situation makes sense for flows dominated by viscosity, for which (27) yields $\tau = O(h^2)$. Using BDF1, the method reads

$$(\delta t \mathbf{D} \mathbf{M}^{-1} \mathbf{G} - \mathbf{S}_{pp}) \mathbf{P}^{n+1} = \delta t \mathbf{D} \mathbf{M}^{-1} (\mathbf{F}^{n+1} - \mathbf{K}(\tilde{\mathbf{U}}_q^{n+1}) \tilde{\mathbf{U}}_q^{n+1}) + \mathbf{D} \mathbf{U}^n - \mathbf{S}_{py} \mathbf{Y}^n$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+1}) \mathbf{U}^{n+1} + \mathbf{G} \mathbf{P}^{n+1} = \mathbf{F}^{n+1}$$

$$\mathbf{M} \mathbf{Y}^{n+1} - \mathbf{G} \mathbf{P}^{n+1} = \mathbf{0}$$

with the notation introduced earlier. Of course, approximation (13) can now be applied. On top of the stability bounds given by Theorems 1–5, we now have the following.

Theorem 6

Assume that $\tau = O(h^2)$. Then, the pressure satisfies the stability estimate:

$$\sum_{n=1}^N \delta t \|h \nabla p_h^n\|_0 \leq C$$

8. NUMERICAL TESTS

In this section we present some numerical results to test the VC and predictor–corrector schemes designed in this paper. In all cases we use approximation (13). Our reference will be the solution of the monolithic problem, that is, the coupled calculation of the velocity and the pressure. We also compare the behaviour of these methods with the stabilized PC and predictor–corrector systems studied in [6, 9], another kind of pressure segregation methods with a similar computational cost. The results shown along this section are obtained with the stabilized versions of the methods using OSS introduced in Section 5. This has allowed us to use equal velocity–pressure interpolation. In particular, we have considered linear elements. Both for PC and VC methods only second-order splitting error methods have been analysed, due to their clear superiority in comparison to first-order methods (see [9]) without significant additional computational cost. The only possible

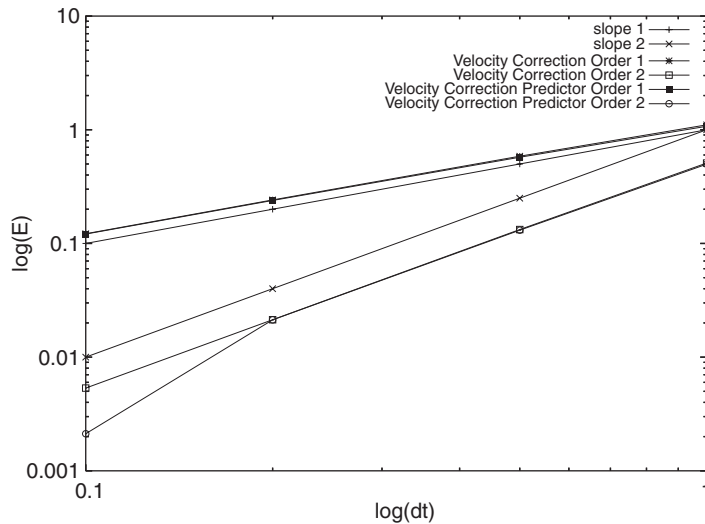


Figure 1. Convergence test.

reason that would (hardly) justify PC methods with a first-order splitting error is their inherent stability. However, this inherent stability is weaker in the case of VC. In any case, the introduction of consistent stabilization techniques, such as OSS, makes this stability unnecessary, improving the accuracy.

The numerical examples presented in this section are the same as in [9]. Likewise, very similar comments apply.

8.1. Convergence test

The first example we consider is a *simple convergence test whose goal is to check numerically the rate of convergence in time for some of the numerical methods described*.

The computational domain is the unit square, discretized using a uniform triangular mesh of 11×11 nodal points (200 triangles). The boundary and initial conditions and the force term are prescribed so that the analytic solution is $\mathbf{u} = (y, -x) \sin(\pi t/10) \exp(t/25)$ and $p = 0$. Note that the exact solution belongs to the finite element space, and thus the only source of numerical error is the time approximation.

Results are shown in Figure 1. The error E is measured in the ℓ^2 norm of the sequence $\{\mathbf{u}^n - \mathbf{u}(t^n)\}$. It is seen that all the methods show the expected rate of convergence. This is particularly relevant for the predictor–corrector schemes, whose error is affected by the convergence tolerance adopted in the iterative loop of each time step.

8.2. Flow in a cavity

In this second example, we solve the classical cavity flow problem at a Reynolds number $Re = 100$. The computational domain is the unit square, discretized using a mesh of 21×21 nodal points (400 triangles). The velocity is fixed to zero everywhere except on the top boundary, where it is prescribed to $(1, 0)$.

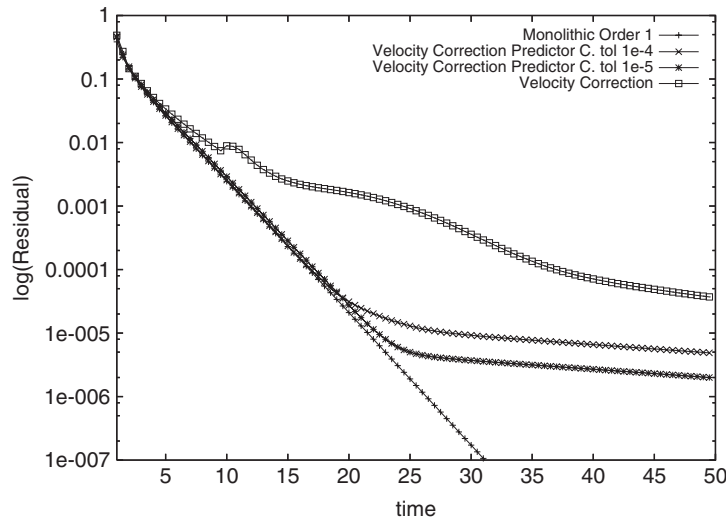


Figure 2. Evolution towards the steady state for the cavity flow problem using monolithic and velocity-correction schemes.

Even though the solution in this simple example is stationary, we obtain it by stepping in time. The goal of this test is precisely *to check the properties of the schemes proposed for the long-term time integration* of stationary solutions (very often difficult to obtain in a stationary calculation) *and, particularly, their numerical dissipation*. The time step employed is $\delta t = 1$.

Figure 2 shows the evolution towards the steady-state case for VC methods compared with the monolithic one. We take a VC method with a second-order splitting error ($p = q = 1$). It is observed that the VC methods reach the steady-state case slower than the monolithic system, which is more dissipative.

The VC predictor–corrector method, as expected, follows very closely the plot of the monolithic system. Furthermore, when the residual reaches the tolerance of the predictor–corrector loop, the plots of the VC predictor–corrector methods detaches from the monolithic one, tending slower to the steady-state case, as it could be expected.

On the other hand, the use of the consistent or lumped mass matrix seems that does not affect essentially the results obtained.

In Figure 3, the VC method is compared to its dual PC method. The original fractional step method with, for simplicity, BDF1 for the time integration and a p th order pressure extrapolation is (see [9] for details)

$$\mathbf{M} \frac{1}{\delta t} (\tilde{\mathbf{U}}^{n+1} - \mathbf{U}^n) + \mathbf{K}(\tilde{\mathbf{U}}^{n+1}) \tilde{\mathbf{U}}^{n+1} + \mathbf{G} \tilde{\mathbf{P}}_p^{n+1} = \mathbf{F}^{n+1}$$

$$\delta t \mathbf{L}(\mathbf{P}^{n+1} - \tilde{\mathbf{P}}_p^n) = \delta t \mathbf{D} \tilde{\mathbf{U}}^{n+1}$$

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1} - \tilde{\mathbf{U}}^{n+1}) + \mathbf{G}(\mathbf{P}^{n+1} - \tilde{\mathbf{P}}_p^{n+1}) = 0$$

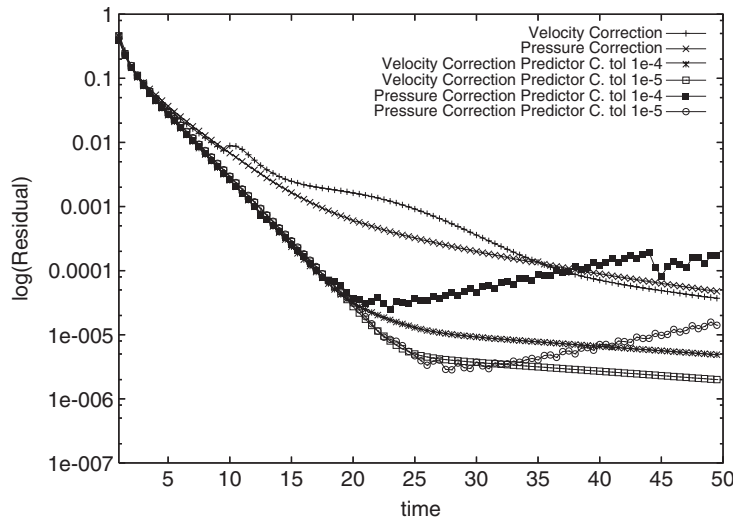


Figure 3. Evolution towards the steady state for the cavity flow problem using velocity and pressure-correction schemes.

whereas the associated predictor–corrector method using a single loop is

$$\mathbf{M} \frac{1}{\delta t} (\mathbf{U}^{n+1,i+1} - \mathbf{U}^n) + \mathbf{K}(\mathbf{U}^{n+1,i}) \mathbf{U}^{n+1,i+1} + \mathbf{G}\mathbf{P}^{n+1,i} = \mathbf{F}^{n+1}$$

$$\delta t \mathbf{L}(\mathbf{P}^{n+1,i+1} - \mathbf{P}^{n+1,i}) = \mathbf{D}(\mathbf{U}^{n+1,i+1})$$

These two methods are the predictor–corrector counterparts of the VC methods given by (16) (with $k = 1$) and (21).

The asymptotic behaviour of both the methods displayed in Figure 3 is very similar. However, their predictor–corrector versions give very different plots. PC predictor–corrector methods have a weird behaviour when tending to the steady-state case. For VC predictor–corrector methods better convergence is attained. The use of a consistent mass matrix or a diagonal mass matrix obtained from a nodal quadrature leads to very similar results. For this reason, only the results obtained with the consistent mass matrix have been included.

Concerning the convergence of the methods, in Figures 4–6 we plot the number of iteration performed *versus* time step in order to compare VC and PC methods (with a maximum of 10 iterations allowed). These iterations deal with the nonlinearity of the convective term for VC and PC methods as well as with the convergence to the monolithic system for predictor–corrector schemes. For this specific test, VC schemes reach convergence slightly faster than PC schemes in all cases. Due to the fact that both PC and VC methods have a very similar computational cost per iteration (in both cases we have to solve a Poisson equation for the pressure and a momentum equation for the velocity) and the CPU time used by both methods is proportional to the number of iterations, VC turns out to be cheaper for this particular example.

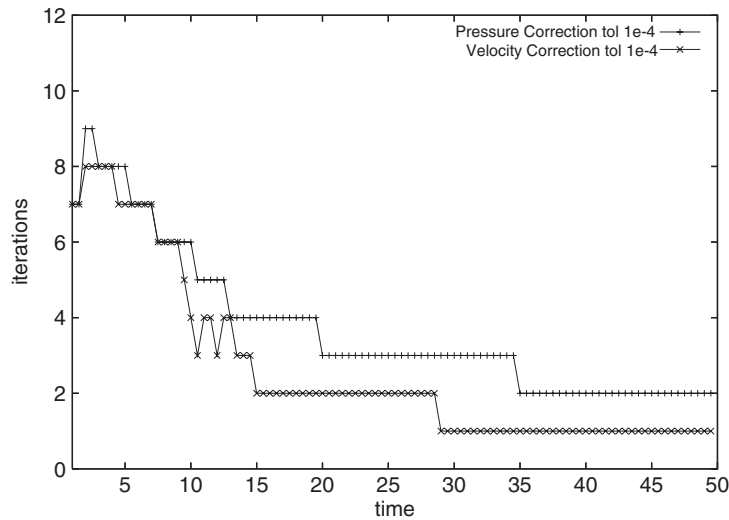


Figure 4. Number of iterations per time step using different velocity and pressure-correction schemes.

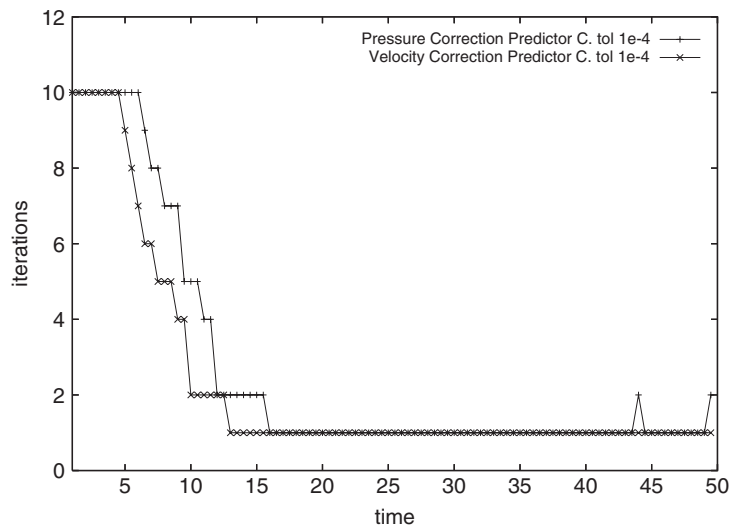


Figure 5. Number of iterations per time step using different velocity and pressure-correction schemes.

8.3. Flow over a cylinder

The last example is also a classical benchmark, namely, the flow over a cylinder. The computational domain is $\bar{\Omega} = [0, 16] \times [0, 8] \setminus D$, with the cylinder D of diameter 1 and centred at $(4, 4)$. The velocity at $x = 0$ is prescribed to $(1, 0)$, whereas at $y = 0$ and 8 the y -velocity component is prescribed to 0 and the x -component is left free. The outflow (where both the x - and y -components

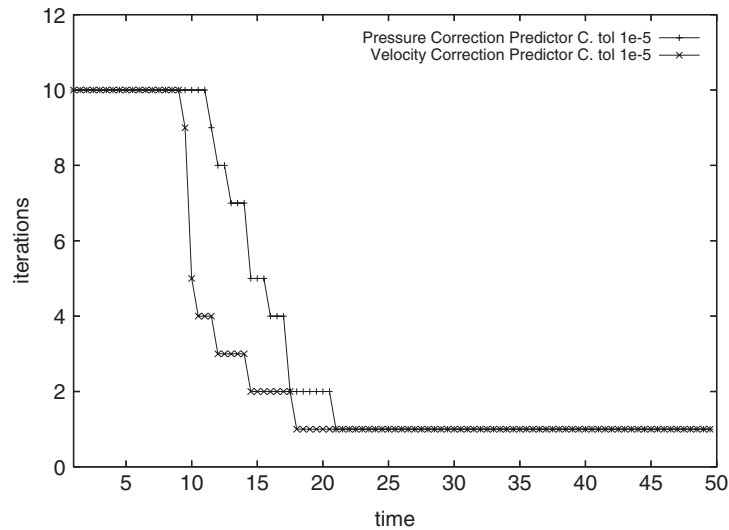


Figure 6. Number of iterations per time step using different velocity and pressure-correction schemes.

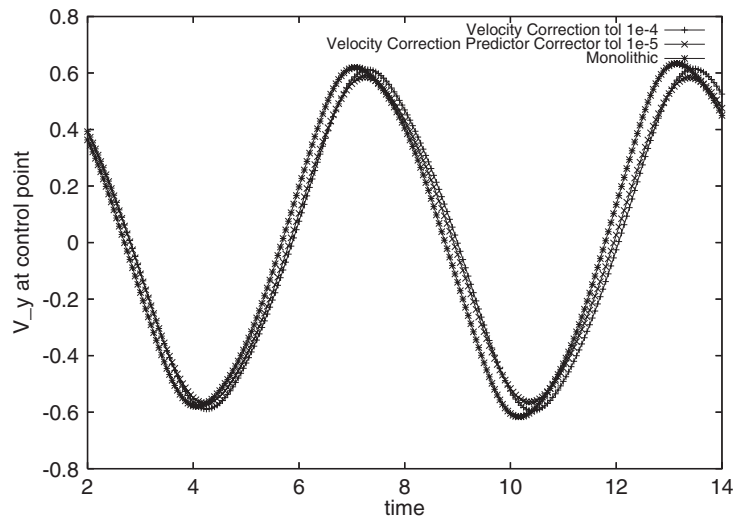


Figure 7. Temporal evolution of the y -velocity component at the control point.

are free) is $x = 16$. The Reynolds number is 100, based on the cylinder diameter and the prescribed inflow velocity. The finite element mesh employed consists of 3604 linear triangles, with 1902 nodal points. This example has been used in [9] to analyse PC methods in time. Here, we try to show the same case with VC methods. Only second-order schemes have been considered.

The evolution of the y -velocity component at the control point located at (6,4) is shown in Figure 7. The time step size used in all the cases is $\delta t = 0.05$. A maximum of only five iterations has been permitted. Similar results to those of the PC methods have been obtained (see [9]). Again, the predictor–corrector method lays between the monolithic and VC method.

9. CONCLUSIONS

The new types of velocity-correction and predictor–corrector methods presented in this paper and obtained at the discrete level from a DPPE monolithic system are appropriate methods for the simulation of flow problems, avoiding the problems that arise when using a pressure Poisson equation obtained at the continuous level. This requires a finite element approximation to $H^2(\Omega)$, which is avoided by our purely algebraic approach.

The velocity-correction methods proposed herein have a behaviour similar to the widely used pressure-correction methods. Moreover, in both cases we observe that third-order methods (in time) become conditionally stable. This is a somehow surprising result, because the perturbation introduced by the splitting is different in both cases.

A nice feature of the DPPE monolithic system is the fact that velocity-correction predictor–corrector schemes arise naturally. We have compared these new schemes to predictor–corrector schemes motivated from pressure-correction methods. Numerical experimentation shows that these new methods exhibit a similar or, in most cases, better behaviour.

From the numerical analysis it seems that the inherent pressure stability of velocity-correction schemes is weaker than the one of pressure-correction methods. Whereas for first-order pressure-correction methods a stability bound for the pressure gradient can be obtained (under some mild assumptions), only part of this term is controlled using velocity-correction methods. Nevertheless, we can recover control over the rest of the pressure gradient from the momentum equation. Using OSS stabilized versions of these schemes have been designed and their enhanced stability properties have been proved.

REFERENCES

1. Gresho PM, Sani RL. *Incompressible Flow and the Finite Element Method*. Wiley: New York, 2000.
2. Chorin AJ. A numerical method for solving incompressible viscous problems. *Journal of Computational Physics* 1967; **2**:12–26.
3. Temam R. Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires (I). *Archives for Rational Mechanics and Analysis* 1969; **32**:135–153.
4. van Kan J. A second-order accurate pressure correction scheme for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**:870–891.
5. Guermond JL, Shen J. Velocity-correction projection methods for incompressible flows. *SIAM Journal on Numerical Analysis* 2003; **41**:112–134.
6. Codina R. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics* 2001; **170**:112–140.
7. Codina R, Soto O. Approximation of the incompressible Navier–Stokes equations using orthogonal-subscale stabilization and pressure segregation on anisotropic finite element meshes. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:1403–1419.
8. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:4295–4321.
9. Codina R, Badia S. On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:2900–2918.

10. Hundsdorfer W, Verwer JG. *Numerical Solution of Time-dependent Advection–Diffusion–Reaction Equations*. Springer: Berlin, 2003.
11. Shen J. A remark on the projection-3 method. *International Journal for Numerical Methods in Fluids* 1993; **16**:249–253.
12. Perot JB. An analysis of the fractional step method. *Journal of Computational Physics* 1993; **108**:51–58.
13. Quarteroni A, Saleri F, Veneziani A. Factorization methods for the numerical approximation of Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2000; **188**:505–526.
14. Blasco J, Codina R. Space, time error estimates for a first order, pressure stabilized finite element method for the incompressible Navier–Stokes equations. *Applied Numerical Mathematics* 2001; **38**:475–497.
15. Codina R, Blasco J. Analysis of a stabilized finite element approximation of the transient convection–diffusion–reaction equation using orthogonal subscales. *Computing and Visualization in Science* 2002; **4**:167–174.
16. Heywood JG, Rannacher R. Finite element approximation of the nonstationary Navier–Stokes problem. IV: error analysis for second-order time discretization. *SIAM Journal on Numerical Analysis* 1990; **27**:353–384.
17. Brenner SC, Scott LR. *The Mathematical Theory of Finite Element Methods*. Springer: Berlin, 1994.